COBB

INSIDE DOS®™

Tips & techniques for MS-DOS & PC-DOS

December 1992
Vol. 3 No. 12

RAM disks: not quite the best of both worlds

Since version 3.2, DOS has included a special driver, RAMDRIVE.SYS, that allows you to create a RAM disk, or "virtual disk," in your computer's random access memory (RAM). Many DOS users shy away from this capability, and even experts disagree over how best to use RAM disks.

Over the years, *Inside DOS* subscribers have asked us a number of questions about RAM disks. In this article, we'll attempt to answer those questions. We'll look at both the advantages and disadvantages of using a RAM disk. Then, we'll show you how to create your own RAM disk. A related article on page 6, "Placing Temporary Files on a RAM Disk Can Speed Up Some Applications," shows you one of the best uses for a RAM disk. Let's get started by defining a RAM disk and looking at some ways to use one.

RAM disk basics

As its name implies, a RAM disk is a portion of random access memory that you set aside to act as a disk drive. You tell DOS to create a RAM disk by placing a DEVICE command in your CONFIG.SYS file. (We'll explain more about this command later.) DOS will assign the RAM disk the next highest drive letter. For example, if you have one hard disk drive named C:, your RAM disk will become the D: drive.

In effect, a RAM disk combines the speed of RAM with the file-management features of a hard disk. Since a RAM disk provides an electronic, rather than a mechanical, storage area, it can access information much faster than a hard disk (10 to 50 times faster, by some estimates). Because of this ability, a RAM disk can significantly cut the time it takes your computer to do *some* disk-intensive tasks. And, like a hard disk, it allows you to create directories and store files—but only *temporarily*. When you reboot or turn off your PC—or when the power fails—all the information stored in your RAM disk will disappear.

Besides their temporary nature, RAM disks have another significant drawback: They use a portion of the random access memory that programs need in order to run properly. If you have only 1 Mb of RAM, you'll probably find that your applications won't run properly when you try to use a RAM disk. However, if you have 2 Mb of memory or more, you might find that a RAM disk boosts your system's performance.

Uses for a RAM disk

RAM disks are an undeniably controversial topic among DOS users. Some contend that using at least one RAM disk is essential to getting the best performance out of your PC. Others find that the memory a RAM disk requires leaves too little for the other applications the user needs. To make matters more confusing, your hardware and software will determine how effective a RAM disk is, or if it will work for you at all. With these caveats in mind, let's look at some uses for RAM disks.

Some DOS users place frequently accessed files on a RAM disk. These files might include database files, spell-checkers or dictionary files, even large document and spreadsheet files that you need to edit. However, copying data files to a RAM disk has fallen out of favor in recent years. Since most applications are now able to load into RAM most or all of the file you want to edit, you probably wouldn't notice much of a speed increase by placing a data file on a RAM disk. More important, you risk losing any information you've entered into the file since you copied it to the RAM disk. For example, if you update a spreadsheet that you've copied to the

IN THIS ISSUE

RAM disk, you'd have to copy it back to your hard disk drive *before* you turned off your PC or rebooted. Otherwise, you would lose all of the changes you made to your RAM disk copy. Similarly, if you add words to a dictionary placed on a RAM disk, you'll have to copy the updated dictionary back to your hard disk drive to save the new words. Unless you have a painfully slow hard disk drive, the risks of placing data files on a RAM disk outweigh the benefits.

Of course, every rule has an exception: A relatively new use for RAM disks is to conserve power on battery-operated laptop computers. Reading from and writing to a mechanical disk drive require much more energy than reading from and writing to RAM. If you want to extend battery life and you have plenty of extended memory, you may want to try copying files you need to edit to a RAM disk. (Remember, you'll need to copy the files back to your hard disk *before* you turn off your laptop.)

You also can place executable program files on a RAM disk, but this use has also diminished. The technique allowed people with dual floppy disk systems to avoid switching diskettes whenever DOS needed to access the program file.

As you can see, using a RAM disk for files you need to preserve is a slightly dangerous proposition. But placing *temporary* files on a RAM disk is a relatively safe technique that can speed up your system. We describe this use for a RAM disk in the article on page 6.

RAM disks vs. disk caches

Since RAM disks and disk caches both place information from the disk into memory, many people wonder which device is more effective. In general, disk caches are more versatile and easier to use than RAM disks. You can install a disk cache, such as DOS' SMARTDRV.SYS, simply by placing a DEVICE statement in your CONFIG.SYS file. But if you're using a RAM disk, the DEVICE statement is only a first step. You then must copy files to the RAM disk for it to be useful. If you change files on the RAM disk, you also must copy the files back to the hard disk.

Disk caches also are "smarter" than RAM disks. Caches track the information accessed from the disk, and place into RAM the information you're most likely to ask for. With a RAM disk, you have to make those decisions.

Creating a RAM disk

Now, after considering the advantages and disadvantages of RAM disks, you might be ready to try creating one. You do so by installing a special device driver,

INSIDE DOS™

Inside DOS (ISSN 1049-5320) is published monthly by The Cobb Group.

Prices

Domestic	$49/yr.	($6.00 each)
Outside U.S.	$69/yr.	($8.50 each)

Phone

Toll free	(800) 223-8720
Local	(502) 491-1900
Customer Relations FAX	(502) 491-8050
Editorial Department FAX	(502) 491-4200

Address

You may address tips, special requests, and other correspondence to:

The Editor, *Inside DOS*
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

For subscriptions, fulfillment questions, and requests for bulk orders, address your letters to:

Customer Relations
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

Postmaster

Second class postage is paid in Louisville, KY. Send address changes to:

Inside DOS
P.O. Box 35160
Louisville, KY 40232

Back Issues

To order back issues, call Customer Relations at (800) 223-8720. Back issues cost $6.00 each, $8.50 outside the U.S., and you can pay with MasterCard, VISA, Discover, or American Express, or we can bill you. Please identify the issue you want by the month and year it was published. Customer Relations can also provide you with an issue-by-issue listing of all the articles that have appeared in *Inside DOS.*

Staff

Editor-in-Chief	Suzanne Thornberry
Contributing Editors	Van Wolverton David Reid
Editing	Linda Watkins Polly Blakemore
Production	Margueriete Stith
Design	Karl Feige
Publications Manager	Elayne Noltemeyer
Managing Editor	Jim Welp
Circulation Manager	Brent Shean
Publications Director	Linda Baughman
Editorial Director	Jeff Yocom
Publisher	Douglas Cobb

RAMDRIVE.SYS, through your CONFIG.SYS file. (This file is called VDISK.SYS in some brands of PC-DOS.) DOS then will create the RAM disk when you boot up and assign it the next available driver letter automatically. For example, if you have one disk drive named C: on your system, DOS will name the RAM disk you create D:. (In unusual circumstances, you may need to add an additional command to your CONFIG.SYS file so that DOS can recognize the RAM disk. The article "A Note on LASTDRIVE" on page 4 explains this situation.) Once you've created the RAM disk, you can create directories on it, copy files to or from it, and even run CHKDSK on it.

You can add the RAMDRIVE command to your CONFIG.SYS using the DOS Editor or any word processor that allows you to create ASCII (text-only) files. You'll want to place the statement that creates the RAM disk *after* any statements you might use to manage extended or expanded memory. To be safe, you can simply place the RAMDRIVE statement at the end of your CONFIG.SYS file. The basic form of the statement is

```
device=c:\dos\ramdrive.sys
```

This statement creates a RAM disk in conventional memory. By default, the RAM disk is 64 Kb, divided into sectors of 512 bytes each, and can hold 64 file entries. If you'd like, you can adjust these values by adding optional information to the statement that installs the RAM drive. You place the additional information in the order shown below:

```
device=c:\dos\ramdrive.sys [disksize] [sectorsize] [entries]
```

At first glance, these specifications might seem confusing. Let's take a close look at what these specifications mean and how you decide on setting them.

Disksize, as you'd expect, is the size (in kilobytes) of the RAM disk itself. If you have only conventional memory, you won't be able to create a very large RAM disk. The default size of 64 Kb may be the upper limit for your RAM disk. But, if you have more than 1 Mb of memory, you can create a larger RAM disk. Theoretically, you can specify any size from 16 to 32767 Kb—as long as you have enough RAM. (If you're using HIMEM.SYS, the limit is 16 Kb.) For example, you can create a 128 Kb RAM disk in conventional memory with the following statement:

```
device=c:\dos\ramdrive.sys 128
```

(As you'll see in a moment, you may be able to create a larger RAM disk *within* extended or expanded memory.)

Sectorsize tells DOS how to allocate the space in the RAM disk. If you plan to place just a few large files on the RAM disk, you'd theoretically be better off with the larger sectors provided by the default of 512 bytes. In fact, Microsoft recommends that you accept the default sector size in almost all circumstances. However, if you plan to place a lot of small files on the RAM disk, you might opt for a sector size of 128 bytes—the smallest sector size available from RAMDRIVE.SYS. Or, you can compromise by selecting a medium setting of 256 bytes. If you want to select either 128 or 256 bytes for the sector size, you must first specify the size of the RAM disk, even if it's the default size of 64 Kb. For example, you can create a 64 Kb RAM disk with 128-byte sectors by placing the following command in your CONFIG.SYS file:

```
device=c:\dos\ramdrive.sys 64 128
```

Entries is the maximum number of directory entries you can create on the root directory of the RAM disk. Each directory or file you create is one directory entry, and the volume label of the RAM disk drive uses one more directory entry. (DOS requires 32 bytes for each directory entry.) The default setting for *entries* is 64, but you can choose any number from 2 to 1024, although RAMDRIVE.SYS may round up the number in order to make the best use of the sector size you've chosen. When you expect to place a lot of files on the RAM disk, you may want to specify a number larger than the default of 64. On the other hand, you can save a little space by choosing a lower number for *entries*—if you're sure you won't need to place a lot of files in your RAM disk. For example, suppose you want to place only a 122 Kb spell-checker on the RAM disk. First, you'll need to choose a size slightly larger than the spell-checker file. Then, you can choose a large sector size (since you're only copying one large file to the RAM disk) and an *entries* setting of 2:

```
device=c:\dos\ramdrive.sys 128 512 2
```

When you reboot after making this change, you'll see a message from RAMDRIVE.SYS telling you that the number of directory entries is 16. Since DOS has to use 512 bytes in a sector, it figures it can use the whole sector for directory entries. At 32 bytes each, that means it can fit 16 entries in the sector. Although you can save a little space in the RAM disk by choosing a lower number, you'll have to choose a larger value for entries if you want to use the RAM disk for more than 64 files.

Creating a RAM disk in extended or expanded memory

As we mentioned earlier, you'll need either extended or expanded memory to create a very large RAM disk. In theory, you could create a RAM disk that used all but 64 Kb of your available memory. In practice, however, you'd probably receive a lot of *Insufficient memory* errors if you tried to create a RAM disk that consumed even 25 percent of your available memory (or, very roughly, 128 Kb from a PC with 640 Kb of conventional memory).

If you have extended or expanded memory, you'll be able to create the RAM disk there, saving your conventional memory for running programs. (If you want to create the RAM disk in extended or expanded memory, you *must* place the RAMDRIVE statement *after* the statements installing your memory managers, such as HIMEM.SYS and EMM386.EXE.) You use the *disksize*, *sectorsize*, and *entries* settings as we described earlier. Then, to specify that DOS create the RAM disk in extended memory, you simply add the /E switch to the end of the RAMDRIVE statement.

For example, suppose you want to create a 1024 Kb (or 1 Mb) RAM disk in extended memory, and you want to use the default values of 512 bytes for the sector size and 64 for the maximum number of file entries. You can create this RAM disk by placing the following statement in your CONFIG.SYS file:

```
device=c:\dos\ramdrive.sys 1024 /e
```

You can use almost the same command to create a similar RAM disk in expanded memory. However, instead of using the /E switch, you use the /A switch to specify expanded memory, as we've done below:

```
device=c:\dos\ramdrive.sys 1024 /a
```

Note, however, that RAM disks work faster in extended memory than in *simulated* expanded memory. Since DOS' EMM386.EXE driver simulates expanded memory, you'd be better off using the /E switch to specify extended memory (managed by HIMEM.SYS) for the RAM disk.

Using the RAM disk

When you first boot up, your RAM disk contains nothing. Each time you start a session, you'll have to copy the files you want to work with onto the RAM disk. You'll also have to create any directories you need for the RAM disk.

Fortunately, you use the same commands to copy files and make directories on the RAM disk as you would on a hard disk. For example, suppose you want to copy the BUDGET.XLS spreadsheet from your C:\EXCEL\DATA directory to the root directory of your RAM disk. To do this, you simply type

```
C:\>copy c:\excel\data\budget.xls D:
```

When you press [Enter], DOS will copy the file. Remember, when you start Excel, you'll have to specify that it load the copy of BUDGET.XLS on the D: drive.

Saving files from the RAM disk

As we said earlier, you must be careful when you place data files on the RAM disk because you almost always want to save any changes you've made. (Usually,

A note on LASTDRIVE

As we noted in the preceding article, "RAM Disks: not Quite the Best of Both Worlds," DOS will assign the RAM disk the next available drive letter. However, you may wonder if you need to use the LASTDRIVE configuration command, which Van Wolverton discussed briefly in his November 1992 column. As Van noted, the LASTDRIVE command specifies the highest drive letter DOS will recognize.

Most DOS users won't need to use the LASTDRIVE command when they create a RAM disk. That's because the DOS 5 default setting for LASTDRIVE is always one drive higher than the last drive currently assigned. But if you use additional drives (such as CD-ROM drives) or use the SUBST command, you may need to use a LASTDRIVE command when you use a RAM disk.

Even if you've partitioned your hard drive into several logical drives, DOS will still be able to create a RAM disk. (By the way, partitioning your hard drive is a very radical process, and it's usually done when you first set up your computer. You can't change your partitions without losing all of the data on your hard disk drive.) For example, suppose you partitioned your 60 Mb hard disk into three 20 Mb logical drives: C:, D:, and E:. In this situation, DOS would set the LASTDRIVE to F:, allowing you to create and use a RAM disk known as the F: drive.

In our tests, DOS was even able to create two RAM disks. For example, suppose you have a single-partition C: drive and you place the following lines in your CONFIG.SYS file:

```
device=c:\dos\ramdrive.sys 128
device=c:\dos\ramdrive.sys 512
```

When you reboot, DOS will create a 128 Kb RAM disk designated as D:, and a 512 Kb RAM disk designated as E:. After creating the D: RAM disk, DOS automatically sets the LASTDRIVE limit to E:, allowing it to create the second RAM disk.

program files don't have to be copied back to the hard disk, since you don't change them by running them from the RAM disk.) Because the RAM disk is obliterated when you reboot or turn off your PC, you'll need to copy any updated files back to a diskette or to the hard drive after you've finished editing them. Of course, you can do this manually by issuing the COPY command.

Returning to our example, suppose you finish updating the copy of the BUDGET.XLS spreadsheet on your RAM disk (D:). After you quit Excel, you can copy the updated version of the file over the old version with the following command:

```
C:\>copy d:\budget.xls c:\excel\data
```

DOS will overwrite the old version of the file C:\EXCEL\DATA\BUDGET.XLS with the version on your RAM disk. Your updated file will be stored on your hard disk, so it's safe to turn off your PC.

Note

As we've said, the optimal settings for your RAM disk depend on both your hardware and the applications you're using. You may have to experiment with the configuration to find what works best for you. For example, if you receive *Insufficient memory* messages when you try to run an application in conventional memory, you should try reducing the amount of memory the RAM disk uses. On the other hand, you might run out of room on your RAM disk for the files you need. In that case, you can try enlarging the RAM disk by specifying a larger disk size. (Generally, only people who create the disk in extended or expanded memory have this luxury.)

Conclusion

While a RAM disk is not a magical solution for pokey applications, using a RAM disk can allow your PC to do some things more quickly. In this article, we've presented the basics for creating and using a RAM disk. If you want to try using a RAM disk to improve your system's performance, you'll probably need to fine-tune the RAM disk's settings.

The following articles discuss other aspects of your system's random access memory:

"Understanding Your Computer's Memory," July 1992

"Creating a Disk Cache with SMARTDrive Can Speed Up Your System," July 1992

"The CONFIG.SYS File Helps You Get the Most from Your System," November 1992

For information on ordering back issues, see the masthead on page 2.

Setting LASTDRIVE

As you can see, your system will normally handle RAM disks quite well by using DOS' default LASTDRIVE setting. However, adding a RAM disk can sometimes prevent DOS from recognizing another drive you've set up. In that case, changing your system's LASTDRIVE command will probably solve the problem.

For example, adding a RAM disk will prevent DOS from recognizing "drives" you've set up with the SUBST command. As we explained in the article "Two Ways to Shorten Your Command Path," which appeared in the March 1992 issue of *Inside DOS*, the SUBST command lets you assign a drive letter to a particular directory on your hard disk drive. After adding the RAMDRIVE statement to your CONFIG.SYS file, you may find that some of your SUBST commands no longer work. If this happens, simply add a LASTDRIVE statement to your CONFIG.SYS. You'll want to choose the highest drive letter you expect to use. Be sure to count:

- each logical partition on your hard disk
- each RAM disk you create
- each SUBST command you want to use

Suppose you have a single-partition C: drive and one RAM disk, known as the D: drive. Let's also suppose you want to use the following SUBST commands:

```
C:\>subst e: c:\word
C:\>subst f: c:\games
```

In this situation, you'd need to add the following line to your CONFIG.SYS file:

```
lastdrive=f
```

When you reboot, DOS will be able to create your RAM disk and make the substitutions you specified for the E: and F: drives.

You also may need to increase your LASTDRIVE setting if you have a special device, such as a CD-ROM drive, that DOS assigns a drive letter. Simply add a LASTDRIVE command to your CONFIG.SYS file or edit the one that's already there to reflect the highest drive letter you'll need to use. However, if your PC is on a network, consult your network administrator before changing the LASTDRIVE setting.

PERFORMANCE TIP

Placing temporary files on a RAM disk can speed up some applications

In the accompanying article, "RAM Disks: not Quite the Best of Both Worlds," we show you how to create a RAM disk and consider the advantages and disadvantages of using one. In this article, we'll highlight what many DOS experts call the best use of a RAM disk: holding temporary files.

Many applications—including certain DOS commands and the DOS Shell—create temporary files. Usually, the command or program will create these temporary files on your hard disk drive, then delete them after the files have served their purpose. However, the hard disk drive may not be the most efficient location for temporary files. Programs can create, read, and delete files more quickly from a RAM disk than from a hard disk drive. With this in mind, we'll begin a brief review of temporary files.

How programs use temporary files

Temporary files sometimes provide a holding place for data as DOS or another program manipulates it. For example, when you pipe the output of one command into another, DOS creates a temporary version of the file you're processing. If you issue the command

```
C:\DATA>type company.txt | sort
```

DOS will need to create a temporary copy of the file COMPANY.TXT so it can sort the lines of the file into alphabetical order.

Unless you tell it otherwise, DOS will create these temporary files in your C:\DOS directory. You won't see these files listed in the directory, however, because DOS deletes the temporary files as soon as it finishes using them.

Usually, DOS and applications store temporary files in a particular directory on your hard disk drive. In fact, you can even tell DOS what directory to use for temporary files. You simply use the SET command with the TEMP environment variable. For example, some DOS users create a TEMP directory on their hard disk drive. Then, they specify this directory by placing the following line in their AUTOEXEC.BAT file:

```
set temp=c:\temp
```

When you type this line, be careful not to use any extra spaces. As you can see, we've left no spaces around the = sign. Also, be sure not to leave any extra spaces after specifying the directory. For example, we left no spaces after the p in *c:\temp*. Extra spaces in the SET command can prevent some programs from locating the directory.

Sending temporary files to a RAM disk

If you've set up a RAM disk through your CONFIG.SYS file, you may be able to improve your system's performance by telling DOS to send temporary files to the RAM disk, rather than to your hard disk drive. Since temporary files are almost always expendable, they are one of the least risky things to put on a RAM disk. Within applications, you can tell the print spooler to write the temporary files it creates to the RAM disk, and you'll probably notice a slight increase in printing speed when you use the spooler.

You also can send any temporary files DOS creates to the RAM disk. For example, suppose you've created a RAM disk named D:. You can send temporary files to the root directory of the RAM disk with the following SET command:

```
C:\>set temp=d:\
```

In fact, if you decide to use the command frequently, you should place the command *set temp=d:\* in your AUTOEXEC.BAT file.

If you use the RAM disk for other files, such as program or data files, you may prefer to keep temporary files in a separate directory on the RAM disk. Of course, you'll need to create the directory on the RAM disk each time you boot. For example, if you want to create a D:\TEMP file on your RAM disk, you'll need to add the following commands to your AUTOEXEC.BAT file:

```
md d:\temp
set temp=d:\temp
```

An example

Since DOS and applications delete temporary files as soon as they finish using them, you usually won't find temporary files on the RAM disk. However, you can see temporary files created by the DOS Shell if you use its Command Prompt option. Here's how the technique works. From the DOS prompt, type *dosshell* to start the Shell. Next, make sure task swapping is enabled. You can do this by selecting Enable Task Swapper from the Options menu. Then, select the Command Prompt option from the Main program group. The DOS Shell will exit to the DOS prompt. Now, issue the DIR command to see what files are in the TEMP directory you specified. For example, if you set the TEMP directory as D:\, enter the command

```
C:\>dir d:
```

You should be able to see at least two files on the RAM disk. One will be named DOSSHELL.SWP, and the other will have the BAT extension. (We discussed these temporary batch files in the letter "Mysterious Batch Files," which appeared in the April 1992 issue of *Inside DOS*.) For example, these are the files we found on our RAM disk when we applied the technique:

```
5069DOSC  BAT          8   09-25-92   10:18a
DOSSHELL  SWP      35389   09-25-92   10:18a
```

Notes

Your applications may generate error messages, such as *Unable to print*, if they run out of space for temporary files on the RAM disk. If you receive suspicious error messages, try increasing the size of your RAM disk (if you have extended or expanded memory) and increasing the number of *entries*, as we explained on page 3. If this doesn't work, you'll need to specify a TEMP directory on the hard disk drive.

Throughout this article, we've assumed that your applications use the TEMP environment variable to track where they should store temporary files. DOS itself uses the directory specified in the TEMP variable for its temporary files. Microsoft Windows, Lotus 1-2-3 for Windows, and WordPerfect are among the applications that also use this TEMP variable. However, some programs use an environment variable for temporary files named TMP instead of TEMP. Applications that use the TMP environment variable include Microsoft Word and Works and Borland's dBASE.

Since many installation programs add a SET command to your AUTOEXEC.BAT file, you may be able to find out if you have a program that uses the TMP environment variable. You can do this by typing *set* and pressing [Enter] at the DOS prompt. DOS will display all of the settings stored in your environment. If you see a setting that begins with *SET TMP=*, you'll know that some program you've installed uses the TMP variable.

If you have a program that uses the TMP environment variable and you want it to use the RAM disk for temporary files, you can edit the AUTOEXEC.BAT file to set the TMP variable as a directory on your RAM disk, instead of a directory on your hard disk drive. For example, if you want to store temporary files on a RAM disk named D:, add the command *set tmp=d:\* to your AUTOEXEC.BAT file.

Initializing the PRINT command

Contributing editor David Reid submitted this tip.

If you use the PRINT command to print files from the DOS prompt, you've probably faced this message:

```
Name of list device [PRN]:
```

DOS presents this message the first time you issue the PRINT command after turning on your PC or rebooting. Your response must tell DOS where to look for the printer. You can choose from the parallel ports LPT1 and LPT2, or the serial ports COM1 or COM2.

You can avoid ever having to respond to this prompt again by adding a single command to your AUTOEXEC.BAT file. This command simply echoes the name of your print device into the PRINT command. Since this "initializes" the printer, the PRINT command won't prompt you for the setting later.

For example, if your printer is attached to your computer's second parallel port (LPT2), place the following command in your AUTOEXEC.BAT file:

```
echo lpt2 | print
```

The first time you issue the PRINT command after rebooting, DOS will print the files you specify *without* pausing to ask the name of the device.

Since most people use the first parallel port, LPT1, for their printer, DOS assumes that LPT1 and PRN are the same until you tell it otherwise. That means you can use the command

```
echo lpt1 | print
```

or the command

```
echo prn | print
```

to initialize a printer attached to LPT1.

Boot diskettes: a review and a warning

In many issues of *Inside DOS*, we've encouraged you to create a boot diskette. If you're ever unable to boot your computer, a boot diskette will usually allow you to start DOS and begin troubleshooting the startup problem. As you may recall, you can create a boot diskette by placing a blank diskette in your A: drive and issuing the following command:

```
C:\>format a: /s
```

This command formats the diskette, then transfers two hidden system files and the COMMAND.COM file to the A: drive. As Van Wolverton explained in the article "Protecting Your Data and Equipment from Disaster," which appeared in the July 1992 issue, you can copy other files to the boot diskette to make it even more useful. However, simply having a diskette with the essential system files should enable you to access your PC again.

Although we frequently mention that you should have a boot diskette ready for emergencies, some readers have requested more information on this important topic. In this article, we'll briefly review how to create and use a simple boot diskette. We'll also provide some tips for tracking down boot diskette problems.

Boot camp

Before you file away your boot diskette, it's a good idea to test it. Simply place the boot diskette in the A: drive of your PC, then press [Ctrl][Alt][Delete]. (This is known as a "warm boot" because you don't have to turn off your PC.)

You should notice the A: drive light is on longer than usual as the PC reads the system files from your boot diskette. After a few moments, the system should display an A> prompt. (Of course, if you've copied additional files to the diskette, you may have a more customized prompt.)

Troubleshooting the boot diskette

Generally, any diskette you format in your A: drive with the /S switch will be able to boot your PC. But what if your test of the boot diskette fails? Often, it's a problem with the diskette itself. Here are a few possible causes of boot diskette failure:

- *Leaving off the /S switch*—If you leave off this switch, DOS won't transfer the system files your PC needs in order to boot.
- *Using a diskette of the wrong density*—As we explained in the article "Choosing the Right Format Option for the Right Diskette," which appeared in the October 1992 issue of *Inside DOS*, formatting a double-density diskette in a high-density drive leaves you with a diskette of dubious safety and usefulness.
- *Using a damaged diskette*—Although this is rare, diskettes can be imperfect or damaged by magnetism, excessive heat, or pressure.

All of these problems are fairly easy to remedy: Just get a new diskette of the proper density and re-issue the FORMAT command with the /S switch. But, very rarely, a PC won't boot from its A: drive because the diskette drive is not set up to be bootable. We discuss a possible solution to this problem in the letter "Some Zenith PCs May Not Have Bootable A: Drives" on page 12.

Emergency procedures

Suppose the day comes when your computer won't boot up. You see only a blank screen—not even a comforting DOS prompt. In the case of an actual emergency, you'd use your boot diskette just as you did when you tested it. If pressing [Ctrl][Alt][Delete] doesn't reboot the computer, turn off the PC, wait about 10 seconds, then turn it back on. (This process is known as a *cold boot*.) If the PC still won't boot, it probably has a very fundamental problem, such as a flaw in its chips or power supply. You might try booting from another boot diskette, but you'll probably have to call in an expert for service.

In the majority of cases, however, you *will* be able to boot your computer. Then, the challenge is finding out what caused the problem. Since the PC boots from the A: drive, but not from the C: drive, you know that the source of the problem must be on your hard disk. Let's look at a few common causes of boot failure.

Errors in the CONFIG.SYS file

As we've noted in previous articles, you should keep a boot diskette available whenever you edit your CONFIG.SYS or AUTOEXEC.BAT file. Because your system automatically carries out the instructions in these two files whenever it boots, an error in these files can sometimes prevent your system from booting. If you've recently changed either of these files and your PC won't boot, you'll definitely need to begin troubleshooting that file.

The CONFIG.SYS file is much more likely to cause booting problems than the AUTOEXEC.BAT file. The CONFIG.SYS file is potentially troublesome because some configuration commands communicate directly with hardware devices. In general, the DEVICE statements that install HIMEM.SYS and EMM386.EXE are most likely to cause booting problems. (If you install other device drivers with an incorrect command, the CONFIG.SYS file will usually just display an *Error in CONFIG.SYS line x* message, where *x* is the number of the line. You also may see the words *Bad or missing*, followed by the path and name of the file. Although your system will boot anyway, the device causing the error won't work.)

As long as you're able to boot from the A: drive, problems caused by the CONFIG.SYS or problems stemming from the AUTOEXEC.BAT file are easy to remedy. Once you have a DOS prompt, you can change to the C:\DOS directory and use the DOS 5 Editor to edit the file you suspect is causing the problem. For example, to edit the CONFIG.SYS file, you issue the following command:

```
C:\DOS>edit c:\config.sys
```

If you've recently changed the CONFIG.SYS file, a good strategy is to place the REM command and a space at the beginning of the line you've most recently changed. (The REM command "remarks out" the line, telling DOS to ignore it.) Be sure to save the file before you exit to DOS. Next, remove the boot diskette from the A: drive and press [Ctrl][Alt][Delete] to reboot the computer. If you've remarked out the offending line, you should be able to see the usual DOS prompt.

If your PC still won't boot from the C: drive, it's back to the drawing board. You can continue remarking out lines and rebooting. Before re-editing the CONFIG.SYS file, however, you might want to try this simple test. Issue the following command to temporarily rename your CONFIG.SYS file:

```
A>ren c:\config.sys.c:\config.xxx
```

(We've assumed that you've booted off a diskette that doesn't contain a PROMPT command, so we've shown the default A> prompt.) Then, remove the boot diskette and press [Ctrl][Alt][Delete]. If the PC displays a DOS prompt, you'll have narrowed the problem to the CONFIG.SYS file. Then, you can rename your CONFIG.XXX file CONFIG.SYS and try editing other lines in the file.

On the other hand, if the PC still won't boot, you'll know to look for other problems. However, before you continue, rename the CONFIG.XXX file CONFIG.SYS.

COMMAND.COM woes

If you haven't recently changed your CONFIG.SYS file, the next most likely cause of booting problems is the COMMAND.COM file. COMMAND.COM contains the instructions DOS uses to carry out internal commands. (As you may know, the instructions for external commands are contained in the COM and EXE files in your DOS directory. That's why you can issue the DIR command even though there's no DIR.COM file in your DOS directory.)

COMMAND.COM problems are usually easy to diagnose. When you try to boot your PC, DOS will present the message *Bad or missing Command Interpreter*. DOS will display this message after any messages generated by your CONFIG.SYS file.

The solution is simple: Copy the COMMAND.COM file from the boot diskette to the root directory of your C: drive. You can do this with the following command:

```
A>copy a:\command.com c:\
```

Although the solution is easy, it's important to find out why the COMMAND.COM file was bad or missing. If you inadvertently deleted the file from your disk, you don't need to worry. If not, try running CHKDSK to see if you have any bad clusters on your C: drive. Bad clusters on the C: drive probably indicate that the COMMAND.COM file was damaged. If you rule out deletion and a damaged file, your system may have a computer virus.

Viruses

New computer viruses come along every month, so it's nearly impossible for us to tell you what symptoms one might cause on your PC. If you suspect your PC has a virus, it's best to leave the PC alone until you're able to run an anti-virus program. Here are a few widely known anti-virus programs:

- Central Point Software ($129)
  Central Point Anti-Virus 1.4
  15220 N.W. Greenbrier Parkway, Suite 200
  Beaverton, Oregon 97006
  (800) 445-4038

- Fifth Generation Software
  Untouchable 1.1 ($99)
  10049 N. Reiger Road
  Baton Rouge, Louisiana 70809
  (504) 291-7221

- Norton Anti-Virus 2.0 ($99)
  Symantec Corporation
  10201 Torre Avenue
  Cupertino, California 95014
  (800) 441-7234

- Clean-Up ($35), Viruscan ($25), and VShield ($25) (a shareware series)

McAfee Associates
1900 Wyatte Drive, Suite 8
Santa Clara, California 95054
(408) 988-3832

Of course, this is by no means an exhaustive list of effective anti-virus programs. Also, note that many utilities packages, including Central Point's PC Tools and Symantec's Norton Utilities, include anti-virus programs.

Note

If you have an old DOS 3.x or 4.x boot diskette, we recommend that you create a new one for DOS 5. There are two important reasons to upgrade your boot diskettes. First, you'll need the current version of DOS so you can use the external commands stored on your system's hard drive. Second, DOS 3.x can't recognize the larger partitions supported by DOS 5. If you have a partition larger than 32 Mb, you won't be able to use a boot diskette created with the older version of DOS.

DOS BASICS

Getting to know DOS' clock

As you probably know, your computer has a system clock that keeps track of the current date and time. When the clock strikes midnight, the system clock advances to the next day and, if necessary, to the next month and year. Fortunately, the system clock is smart enough to remember how many days are in each month and which years are leap years.

Each time you create or modify a file, DOS pulls the current date and time from your computer's system clock and time-stamps that file. When you use the DIR command to see a directory listing of the current drive, you'll see the date and time each file was last modified, as shown in Figure A. For instance, the first file listed in Figure A, AUTOEXEC.BAT, was last modified on 9-18-92 at 3:39 PM.

Figure A

```
C:\>dir

  Volume in drive C has no label
  Directory of C:\

AUTOEXEC     BAT          260         9-18-92     3:39p
AUTOEXEC     OLD          216         8-12-92    11:05a
CONFIG       SYS          108         6-04-92     2:12p
COMMAND      COM        25276        12-03-87     6:45p
EXCEL              <DIR>             11-19-91    10:27a
WORD               <DIR>              4-02-91     2:47p
```

DOS uses your system clock to time-stamp files and directories.

If you copy a file from a diskette, the file will keep its original time stamp. For example, the COMMAND.COM file shown in Figure A was created on 12-03-87, although we copied it to our hard drive much later when we installed DOS 3.3.

As you can see, DOS also associates a date and time with the directories on your drive. Of course, the date and time associated with a directory indicates when you originally created that directory. In the case of the C:\EXCEL and C:\WORD directories, you can see the date you installed the programs.

The date and time information associated with your files and directories comes in handy in a variety of situations. For instance, if you want to delete the unnecessary data files on a particular disk, you can use the date and time information to determine quickly which files haven't been modified for a while. Or, if you forget the name of a file you were working with earlier that day, you can scan the directory listing for the files that are stamped with the current date.

Setting the date and time

Because DOS uses your computer's system clock to perform a variety of important functions, like time-stamping new files and directories, you'll want to make sure that you keep your system clock updated. Fortunately, the system clock in most computers, including nearly all 286- and 386-based machines, runs all the time. When you turn off the computer, a small battery inside the computer powers the system clock, keeping it updated until you turn on the computer again.

Many older computers, however, including the IBM PC and PC-XT, are not equipped with a battery. If your computer falls into this category, shutting it off will reset its system clock to midnight, January 1, 1980. When you turn the computer back on, it will probably prompt you for the current date and time.

Even if your system clock is powered by a battery, you'll occasionally need to update your date and/or time. For instance, when daylight savings ends or begins, you'll want to adjust your system clock accordingly, if necessary. Fortunately, DOS makes it easy to update your system's current date and time.

To specify the current date, just enter the command

```
C:\>date
```

Immediately, DOS will display the current date and prompt you for the new date:

```
Current date is Tue 12-01-92
Enter new date (mm-dd-yy): _
```

At this point, you can change the current date by typing a new date in mm-dd-yy form. For instance, to change the current date to December 15, 1992, type *12-15-92*. If you want to leave the current date as is, you can simply press [Enter] to return to the DOS prompt.

As you might guess, you can change the current time by issuing the command

```
C:\>time
```

Once you've entered this command, you'll see a prompt like this:

```
Current time is 12:38:43.91
Enter new time: _
```

Now, suppose you need to set your clock back an hour for the fall time change. You can do this by entering *11:38* and pressing [Enter]:

```
Current time is 12:38:43.91
Enter new time: 11:38
```

Of course, if you issue the TIME command and find that your system already has the correct time, leave the *Enter new time* prompt blank and press [Enter].

Note that you don't need to supply all the detail you see in the time prompt—you can just supply the hour and minutes, or just the hour. For example, to specify 9:00 AM, you don't need to enter *9:00:00.00*—you can just enter *9:00* or even *9*.

Entering afternoon and evening times is easier with DOS 5 than it was with previous versions of DOS. You no longer have to use the 24-hour format. Instead, you can enter the abbreviation *a* for AM and *p* for PM. For example, if you want to set the time as 1:32 PM, you can enter *1:32p*. (If you're used to the 24-hour format, you can enter *13:32*.) Perhaps the strangest case of all is setting your clock for midnight—to do this, you simply enter *0*.

LETTERS

A better way to end the SAVER.BAS program

I enjoyed your September 1992 issue, particularly the article on Microsoft QBasic. However, the screen-saver program, SAVER.BAS, lacked one key feature: It offered no way to exit the program and return to DOS without going through the QBasic Editor screen.

You can solve this problem simply by adding the QBasic SYSTEM command to the end of the SAVER.BAS program.

Paul P. Smilgin
Windermere, Florida

Thanks for the excellent suggestion. If you'd like to update your version of SAVER.BAS, simply use the DOS 5 Editor or a word processor that allows you to save an ASCII, or text-only, file. Then, move to the end of the SAVER.BAS file and type *system* on a new line. Figure A shows the improved version of SAVER.BAS. The new SYSTEM command is in color.

When you press a key to stop SAVER.BAS, DOS won't automatically clear the screen. You can further enhance SAVER.BAS by creating a batch file to run QBasic and SAVER.BAS, and then clear the screen. This three-line batch file, SAVER.BAT, is shown in Figure B.

After you've created SAVER.BAT, you can run the screen saver by typing *saver* and pressing [Enter]. When you're ready to continue working with DOS, simply press a key. SAVER.BAS will exit to the DOS environment, which will still be running SAVER.BAT. Therefore, DOS will execute the CLS command at the end of SAVER.BAT.

Figure A

```
' SAVER.BAS--A screen saver written in MS-DOS QBasic.

RANDOMIZE TIMER                 ' seed random number generator

DO                                        ' begin loop
    CLS                                   ' clear screen
    randomColor% = INT(RND * 15) + 1      ' get random color
    row% = INT(RND * 25) + 1              ' get random row
    column% = INT(RND * 72) + 1           ' get random column
    COLOR randomColor%                    ' change color
    LOCATE row%, column%                  ' place cursor
    PRINT TIME$;                          ' display system time
    SLEEP 1                               ' pause 1 second
LOOP WHILE INKEY$ = ""          ' loop until a key is pressed
SYSTEM                                    ' return to DOS
```

SAVER.BAS is a screen-saver program that can extend the life of your monitor.

Figure B

```
@echo off
qbasic /run saver.bas
cls
```

SAVER.BAT will run your screen saver and then clear the screen when you want to return to work.

Microsoft Technical Support
(206) 454-2030

Some Zenith PCs may not have bootable A: drives

I created an emergency system diskette as Van Wolverton described in the article "Protecting Your Data and Equipment from Disaster," which appeared in the July 1992 issue of *Inside DOS*. After editing my CONFIG.SYS file, I was unable to boot from the C: drive of my Zenith 386 computer. I tried booting from the A: drive using my emergency system diskette, but to no avail. I also tried using another boot diskette from a colleague's PC, and was still unable to boot up. Can you offer any suggestions?

David A. Schedler
Richmond, Virginia

According to Microsoft, MS-DOS 5.0 can boot only from the A: or C: drive. However, some systems are not configured with a bootable A: drive. Evidently, this occurs with a number of Zenith PCs, so Microsoft has published a technical note explaining a solution to the problem.

The solution is to access your system's setup program and make the floppy disk bootable. The setup program resides on a chip inside your computer, so you don't need to load DOS before you can change the settings. Immediately after rebooting or turning on your PC, press [Ctrl][Alt][Insert]. This key combination will allow you to access the setup program. Then, enter *BF0* (that's a zero) as a bootable drive. (By the way, your system's C: drive is known as BW0. The W stands for *Winchester*, an old name for a hard disk drive. The F, of course, stands for *floppy*.)

Although this configuration is rather rare, we'd like to remind all DOS users to test your boot diskettes before filing them away. In the article "Boot Diskette: a Review and a Warning," which begins on page 8, we review the basics of creating a boot diskette and provide a few troubleshooting tips.

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION

(Required by 39 U.S.C. 3685). 1A. Title of publication: Inside DOS. 1B. Publication No. 0005265. 2. Date of filing: 9/28/92. 3. Frequency of issue: Monthly. 3A. No. of issues published annually: 12. 3B. Annual subscription price: $49 ($69 foreign). 4. Complete mailing address of known office of publication: The Cobb Group, 9420 Bunsen Pkwy., #300, Lou., KY 40220. 5. Complete mailing address of headquarters of general business offices of the publisher: The Cobb Group, 9420 Bunsen Pkwy., #300, Lou., KY 40220. 6. Full names and complete mailing address of publisher, editor, and managing editor: Publisher, Douglas F. Cobb, The Cobb Group, 9420 Bunsen Pkwy., #300, Lou., KY 40220; Editor, Suzanne Thornberry, The Cobb Group, 9420 Bunsen Pkwy., #300, Lou., KY 40220; Managing Editor, Linda Baughman, The Cobb Group, 9420 Bunsen Pkwy., #300, Lou., KY 40220. 7. Owner: Ziff Communications Co., Ziff Investment Partnership L.P., Philip B. Korsant, the address for all of the foregoing being 1 Park Ave., New York, NY 10016. 8. Known bondholders, mortgagees, and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages or other securities: None. 9. For completion by nonprofit organizations authorized to mail at special rates (DMM Section 423.12 only)—The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes (Check one):__Has not changed during preceding 12 months__Has changed during preceding 12 months. (If changed, publisher must submit explanation of change with this statement.) 10. Extent and nature of circulation—A. Total no. copies (net press run): Average no. copies each issue during preceding 12 months, 81,973; actual no. copies of single issue published nearest to filing date, 57,736. B. Paid and/or requested circulation—1. Sales through dealers and carriers, street vendors and counter sales: Average no. copies each issue during preceding 12 months, 882; actual no. copies of single issue published nearest to filing date, 379. 2. Mail subscription (Paid and/or requested): Average no. copies each issue during preceding 12 months, 69.931; actual no. copies of single issue published nearest to filing date, 52,237. C. Total paid and/or requested circulation: (Sum of 10B1 and 10B2) Average no. copies each issue during preceding 12 months, 70,813; actual no. copies of single issue published nearest to filing date, 52,616. D. Free distribution by mail, carrier or other means. Samples, complimentary, and other free copies: Average no. copies each issue during preceding 12 months, 169; actual no. copies of single issue published nearest to filing date, 164. E. Total distribution (Sum of C and D): Average no. copies each issue during preceding 12 months, 70,982; actual no. copies of single issue published nearest to filing date, 52,780. F. Copies not distributed-1. Office use, left over, unaccounted, spoiled after printing: Average no. copies each issue during preceding 12 months, 10,991; actual no. copies of single issue published nearest to filing date, 4,956. 2. Return from news agents. Average no. copies each issue during preceding 12 months, 0; actual no. copies of single issue published nearest to filing date, 0. G. Total (Sum of E, F1 and 2-should equal net press run shown in A): Average no. copies each issue during preceding 12 months, 81,973; actual no. copies of single issue published nearest to filing date, 57,736. 11. I certify that the statements made by me above are correct and complete. Douglas F. Cobb, Publisher.

CORRECTION

Use ? when renaming a filename containing a space

In November, we published a technique for renaming filenames containing a space. You might need this technique because DOS won't recognize filenames containing a space even though some programs let you include spaces in filenames. Unfortunately, the workaround we published will sometimes generate the following error message:

```
Duplicate file name or file name not found
```

This error occurs because DOS will ignore the remaining portion of the root name of a file when you include the * wildcard. In our example in the November issue, we showed the command *ren my*40 my_40*. But this command would generate the *Duplicate file name* error if you have other filenames beginning with *my* in the same directory. (DOS will rename the first file it finds before generating the error message.)

Fortunately, the problem is easily remedied by using the ? wildcard specification instead of the * wildcard. The ? wildcard tells DOS to substitute exactly one character for the ? symbol. So, the correct form of the command shown in the November issue is

```
C:\>ren my?40 my_40
```

This form of the command will work even if you have other filenames beginning with *my*. We apologize for any inconvenience this may have caused you.

Printed on Halopaque: 100% acid free, made from 50% de-inked, recycled fiber including 10% post-consumer waste. This journal is 100% recyclable.